MAC–IME–USP

Carlos Eduardo Ferreira
Sala 108C Tel.: 3091 6079
e-mail cef@ime.usp.br
monitores Bento Pereira
João Gabriel Lembo

MAC 323 – Algoritmos e Estruturas de Dados II

Primeiro semestre de 2020

Tabelas de símbolos – **Entrega: 13 de abril**

O objetivo deste exercício-programa é testar diferentes implementações para a estrutura de dados **tabela de símbolos ordenada**. Os operadores a serem implementados são:

- `void insere (Chave, chave, Item valor);`
- `Item devolve (Chave chave);`
- `void remove (Chave chave);`
- `int rank (Chave chave);`
- `Chave seleciona(int k);`

A estrutura pode ser implementada utilizando, entre outros:

- vetor desordenado;
- vetor ordenado;
- lista ligada desordenada;
- lista ligada ordenada;
- árvore de busca binária gerada na ordem da entrada;
- treaps;
- árvores 2-3;
- árvores rubro-negras;
- tabelas de hashing.

Neste exercício você deverá implementar de todas as formas acima, e testar empiricamente as implementações com relação ao tempo que as cinco operações definidas acima demoram.

Para fazer seus testes, você deverá implementar uma tabela de símbolos para calcular a frequência que ocorrem palavras em um texto muito grande. Assim, o programa deverá receber na linha de comando o nome de um arquivo de entrada (em formato UTF 8) e construir uma tabela de símbolos com as palavras deste texto. E, para implementar a tabela de símbolos o usuário poderá escolher uma das estruturas listadas acima:

- VD: vetor desordenado
- VO: vetor ordenado
- LD: lista ligada desordenada
- LO: lista ligada ordenada
- AB: árvore de busca binária gerada na ordem de entrada
- TR: treaps
- A23: árvores 2-3
- RN: árvores rubro-negras
- HS: tabelas de hashing

Você poderá utilizar textos grandes, por exemplo, do Projeto Gutemberg:

`http://www.gutenberg.org.`

Neste site você pode fazer o *download* de vários livros que poderá utilizar como entrada em seu EP. Teste com outros tipos de dados, como dicionários, códigos muito grandes, textos gerados aleatoriamente (estilo *loren ipsum* como `https://mussumipsum.com/`), etc.

Depois de montar a tabela de símbolos você deve permitir o uso das operações sobre tabelas de símbolos (inserções, consultas, etc). Os monitores providenciarão o código para esta interface de forma padrão para facilitar os testes.

Junto com seu EP você deverá entregar mais uma vez um **relatório** mostrando os testes que você fez, e as conclusões que obteve. Faça testes com textos de diferentes tamanhos para poder ter uma ideia da complexidade assintótica de cada implementação.

Exemplo de entrada

A chamada do seu programa usará a linha de comando para os parâmetros: nome do arquivo a ser lido e tipo de estrutura usada. Assim, por exemplo, se seu programa se chama `ep1`, a chamada para implementar uma tabela de símbolos usando uma lista ligada ordenada:

```
./ep1 OsLusiadas.txt LO
```

Exemplo de operações

O seu programa vai permitir operações de `insere`, `devolve`, `remove`, `rank` e `seleciona`. Para cada operação serão pedidos os parâmetros e a função correspondente será chamada.

Detalhes da entrega e correção

- Muitos problemas de vazamento de memória podem ser minimizados se vocês usarem o valgrind;
- Os monitores passarão as flags de compilação para compilar este EP.